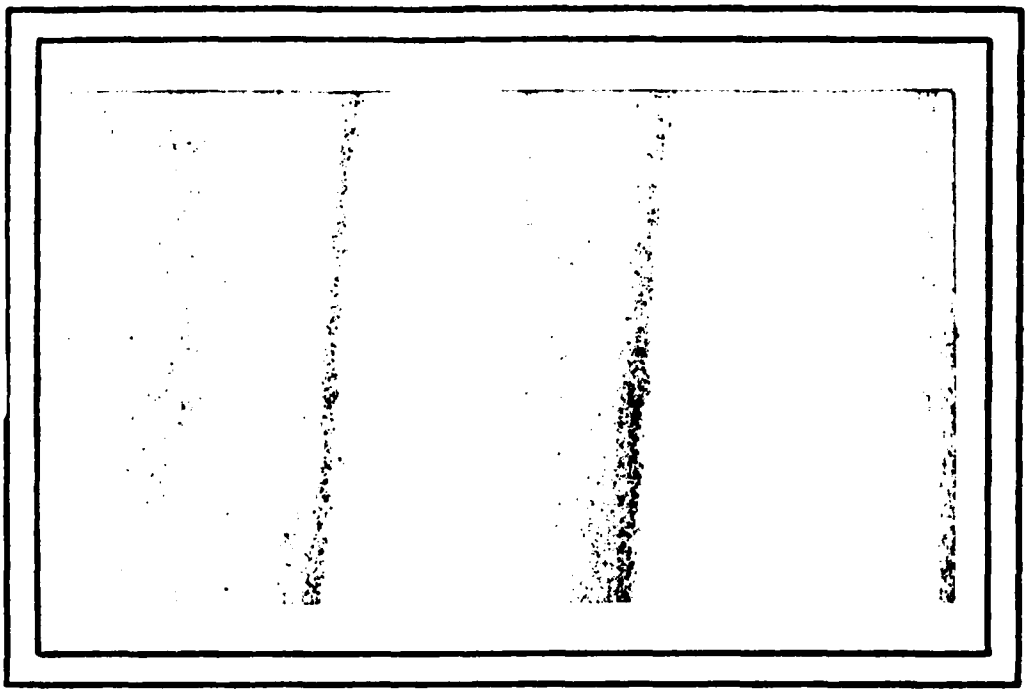


COMPUTER SCIENCE TECHNICAL REPORT SERIES

UNIVERSITY OF MARYLAND
COLLEGE PARK, MARYLAND
20742

TR-1078
DAAG-53-76C-0138

July 1981

FILTERED PROJECTIONS AS AN AID IN CORNER DETECTION

Zhong-Quan Wu*
Azriel Rosenfeld

Computer Vision Laboratory
Computer Science Center
University of Maryland
College Park, MD 20742

ABSTRACT

Corners are very useful features for such purposes as image matching or shape analysis, but corner detection is a relatively expensive operation. This paper uses filtered x and y projections, applied to an image containing an object that has not been explicitly segmented from its background, to determine possible positions of corners, so that corner detection can be applied only in the vicinity of these positions. Even in cases where the object would be hard to segment (unimodal histogram), this approach yields a good set of possible corner positions.

The support of the Defense Advanced Research Projects Agency and the U.S. Army Night Vision Laboratory under Contract DAAG-53-76C-0138 (DARPA Order 3206) is gratefully acknowledged, as is the help of Janet Salzman in preparing this paper. The author also wishes to thank Bob Kirby and Les Kitchen for their help.

*Permanent address: Tsinghua University, Beijing, People's Republic of China

1. Introduction

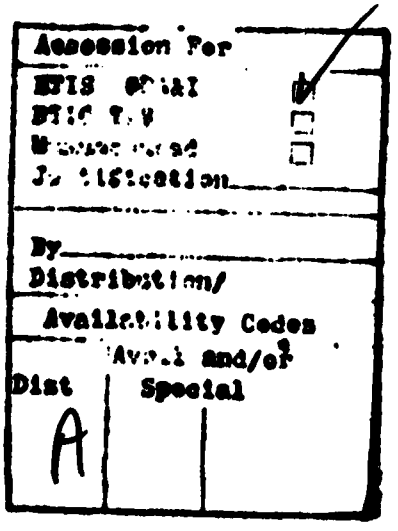
Corner points in an image, at which the gradient magnitude and the rate of change of gradient direction are both high, are useful features for image matching, because they yield sharp matches. The corners of objects that have been extracted from an image are also very useful in shape analysis, e.g., as vertex positions in polygonal approximation. However, corner points in an image are relatively expensive to detect [1,2], since they require computing higher-order difference operators in every position. Corner detection is much less expensive after an object has been extracted from the image; but it may be difficult to extract the object cleanly in order to detect its corners if the image is noisy, e.g., if it has a unimodal histogram.

This paper describes a method of using the x and y projections of an image to detect possible positions in the image where corners may be present. Corners of significant size should give rise to slope discontinuities in the projections, and a filtering process (such as those used in image reconstruction) can be used to produce peaks at these discontinuities. It should be possible to detect discontinuities due to corners even if the image is noisy, since the projection process involves averaging, which reduces the effects of noise.

Each peak on the filtered x(y) projection indicates a column (row) of the image in which a corner may be present, so that the intersections of these rows and columns define possible corner

points. Thus corner detection need be applied only in the vicinity of these points, which are only a small fraction of the points in the image. To further reduce the computational cost, the neighborhoods of the candidate points can be checked for nonuniformity of gray level before the corner detection process is applied to them. Candidate points can also be eliminated by a nonmaximum suppression process.

Section 2 of this paper describes the method and presents results for some images of airplanes and an image of a tank.



2. Method

2.1 Projection, filtering, and peak selection

Given an image $f(x,y)$, its x and y projections are defined respectively by

$$p(x) = \sum_{y} f(x,y) \qquad \text{and} \qquad p(y) = \sum_{x} f(x,y)$$

If we regard the image as the sum of an ideal image plus zero-mean random noise, the projection process reduces the variability of this noise (relative to the signal level in the projection). This may be stated more precisely as follows:

Let $p(x) = \sum_{n_y} f(x,y)$ and $p(y) = \sum_{n_x} f(x,y)$ where n_y is the number of rows and n_x the number of columns. If we regard the image as the sum of an ideal image $f(x,y)$ plus zero mean random noise $z(x,y)$, then this may be rewritten as

$$p_x = \sum_{n_y} (f(x,y) + z(x,y))$$

$$p_y = \sum_{n_x} (f(x,y) + z(x,y))$$

In order to compare p_x and p_y with the pixel gray level, we normalize p_x and p_y:

$$\bar{p}_x = \frac{1}{n_y} p_x = \frac{1}{n_y} \sum_{n_y} f(x,y) + \frac{1}{n_y} \sum_{n_y} z(x,y)$$

$$\bar{p}_y = \frac{1}{n_x} p_y = \frac{1}{n_x} \sum_{n_x} f(x,y) + \frac{1}{n_x} \sum_{n_x} z(x,y)$$

Suppose $z(x,y)$ is a random variable with zero mean and variance σ^2. Then the second term can be regarded as a sample of a random variable with zero mean and variance σ^2/n_y (σ^2/n_x).

Therefore, the projection process reduces the variability of this noise [3].

Straight edges in the image do not give rise to discontinuities in the projections unless they are nearly parallel to one of the axes. Corners in the image, on the other hand, do give rise to slope discontinuities. Thus corners are generally associated with slope changes in the projections, and can also produce discontinuities if one side of the corner lies along an axis. These phenomena are illustrated in Figure 1.

To enhance these (slope) discontinuities and convert them into peaks, we can apply a convolution operation to the projections which acts like a second-difference operator, thus converting slope changes into changes of magnitude. A simple example of such an operation is the one-dimensional digital Laplacian, in which the convolution kernel has value 1 at the origin and value $\frac{1}{2}$ at ± 1. Another example is the Shepp-Logan filter [4], in which the kernel has value approximately 0.2 at ± 1. Note that the Laplacian output is 0 when we convolve it with a linear ramp, whereas the Shepp-Logan filter yields nonzero output (we use a truncated form of this filter).

Figure 2 shows a 74×128 airplane image to which Gaussian noise has been added with signal:noise ratio=5:1*. The histograms of the image is also shown; note that it is quite unimodal.

*s:n is defined here as the ratio between the square of the edge contrast and the variance of the noise; see [5].

Figure 3 shows the x and y projections of the image, and Figure 4 shows the results of filtering the projections of the image by convolution with the Shepp-Logan and Laplacian filters, respectively (a constant has been added to the Laplacian values). Laplacian filtering was used in our experiments, since the Shepp-Logan filter preserves the original values to a great extent, rather than simply responding to changes.

Any peak (=local maximum) on the filtered projection could indicate the presence of a corner in the image, but in order to eliminate peaks due to noise, we use only peaks whose heights are in the top 30% to 50%. The more noisy the image, the smaller the percentage of the peaks that should be accepted.

2.2 Corner selection

Each peak in the x(y) projection indicates the possible presence of a corner in that column (row) of the image. Thus if we have m(n) peaks on the x(y) projection, we have mn positions of possible corners in the image, i.e., the positions where the m columns intersect the n rows, as illustrated in Figure 5. For the airplane image, this yields 150 positions, as seen in Figure 6. Note that in Figure 5 the four corners in the image give rise to four peaks on each projection, and these in turn yield sixteen possible corner positions. Similarly in Figure 6 we have a grid of possible positions for the corners. Thus we have to eliminate most of the possible positions by checking them for the presence of actual corners.

A simple method of eliminating positions at which corners are not present is to test for uniformity of gray level. We use a 3×3 window centered at the given position; if the gray level range r in this window is less than a specified fraction of the average gray level $\bar{g}$ in the window, we call the window uniform. To reduce the effects of noise on the measurement of the gray level range, we can define the range r as the difference between (e.g.) the averages of the three highest and three lowest gray levels in the window, $\bar{g}_{3max}-\bar{g}_{3min}$. Figure 7 shows the results of eliminating positions for which $r<p\bar{g}$, where p=.8,.7, and .6. The results are shown superimposed on the non-noisy airplane image. For this image, p=.7 seems to be the best choice.

When we use p=.7, the number of candidate corner points is reduced to 62; and as Figure 7 shows, these points include many of the major corners on the airplane. Table 1 shows, for each candidate point, its x and y coordinates, its gradient magnitude (G), its rate of change of gradient direction ("turn", T), and the product C=GT, while measures "corner merit".

Figure 8 shows a histogram of the values of C using p=.7. As might be expected, this histogram has a peak at low values of C, and if we threshold C to eliminate this peak, we obtain a selected set of corner points which still includes most of the airplane corners. Figure 9 shows the results of applying various thresholds to C; note that these thresholds eliminate most of the candidate points, including nearly all the candidates that lie in the background.

Another way to eliminate extraneous corner points is by nonmaximum suppression. Figure 10 shows the results of suppressing nonmaxima of C, using a 7×7 neighborhood (i.e., a point is suppressed if there is a point within checkerboard distance 3 with a higher value of C), after thresholding at the three levels. Figure 11 shows the results of nonmaximum suppression after thresholding at 500, using 7×7, 11×11, and 15×15 neighborhoods, corresponding to checkerboard distances 3, 5, and 7. Another possibility is to suppress low values rowwise or columnwise, rather than in a small neighborhood - e.g., to discard the lowest value of C in each row (see starred entries in Table 1). Figure 12

shows the results of doing this (without thresholding) and then suppressing nonmaxima in a 3×3, 7×7, or 11×11 neighborhood.

All these methods of picking good corner points give us a mixture of actual airplane corners and noise corners. In a noisy image, it is not possible to extract the correct set of corners by local processing alone, even with the aid of projections, since the noise often weakens real corners while producing strong false corners. However, our results have a good intersection with the set of correct corners, and if we know what shape we are looking for, we can use point pattern matching techniques [6,7] to determine that a subset of the correct corners is present.

Figure 13 shows another airplane example: (a) airport scene; (b) airplane window and its projections; (c) possible corner positions and result of selecting nonuniform positions; (d) histogram of corner merit values for candidate points; (e) results of thresholding and of nonmaximum suppression; (f) results of nonmaximum suppression without thresholding. As in the first airplane example, good sets of corner points are obtained using these techniques. Figure 14 shows that similar results can be obtained without using projections, by applying the nonuniformity test at every point of the image and thresholding or nonmaximum-suppressing the corner merits; but this is considerably more expensive (see Section 3); corner merit must be computed at the hundreds of points that satisfy the nonuniformity criterion.

Figure 15 shows another example using a noisy infrared image of a tank. Here the nonuniformity criterion (p) is smaller because the image has lower contrast. Our methods yield less than 20 corners, most of which lie on the border of the tank.

3. Computational cost and concluding remarks

Our method compares quite favorably in computational cost with performing corner detection at every pixel. If α is the fraction of rows and columns for which peaks are detected on the projections, then the fraction of pixels that are candidate corner points is α^2; e.g., α=25% implies that only about 6% of the pixels are candidates. Moreover, the uniformity test, which tends to eliminate a majority of the candidates, is much cheaper than the corner detection operator; the latter involves over 50 multiplications, and a similar number of additions, for each pixel. The overhead of computing and filtering the projections is not large; computing them costs about 2 additions per pixel, and filtering costs on the order of the square root of the number of pixels.

We have seen that corner detection in a noisy image is not a very reliable process. Using projections, however, in which the averaging helps to combat the effects of the noise, we can eliminate most of the non-corner points without losing more than a fraction of the corner points. We can further reduce the set of candidate points by applying simple tests for gray level nonuniformity. When we apply corner detection to the remaining candidates, and use thresholding, nonmaximum suppression, or both, we obtain a set of corners comparable to the set that could be obtained by applying corner detection in the image, and at much lower computational cost.

References

1. P. R. Beaudet, Rotationally invariant image operators, Proc. 4th Intl. Joint Conf. on Pattern Recognition, November 1978, 579-583.

2. L. Kitchen and A. Rosenfeld, Gray-level corner detection, TR-887, Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, MD, April 1980.

3. A. Rosenfeld and A. C. Kak, Digital Picture Processing, Academic Press, New York, 1976, p. 34 and p. 197.

4. L. A. Shepp and B. F. Logan,"The Fourier reconstruction of a head section," IEEE Trans. Nuclear Science 21, 1974, pp. 21-42.

5. W. K. Pratt, Digital Image Processing, John Wiley & Sons, New York, 1978, pp. 495-501.

6. D. J. Kahl, A. Danker, and A. Rosenfeld, Some experiments in point pattern matching, IEEE Trans. Systems, Man, Cybernetics 10, 1980, 105-116.

7. S. Ranade and A. Rosenfeld, Point pattern matching by relaxation, Pattern Recognition 12, 1980, 268-275.

X	Y	T	G	C	
32	10	302	6.7	2023	
37	10	-44	18.8	827	*
10	13	-58	12.1	701	*
37	13	206	7.9	1627	
10	27	56	14.9	834	*
37	27	-250	4.3	1075	
37	29	-326	2.3	749	
34	35	21	12.9	270	
37	35	-22	8.6	189	*
32	42	124	17.8	2207	
34	42	28	4.9	137	*
37	42	-28	12.6	352	
32	49	-1072	1.2	1286	
34	49	-98	7.8	764	
37	49	-108	2.0	216	*
42	49	-207	3.8	786	
51	49	-198	2.3	455	
22	60	-77	14.2	1093	
32	60	-109	15.3	1667	
34	60	-360	1.4	504	*
37	60	-120	6.9	828	
42	60	-242	5.0	1210	
51	60	40	18.5	740	
17	66	38	17.5	665	
22	66	50	11.0	550	
32	66	-21	4.3	90	*
34	66	-296	3.7	1095	
37	66	-318	6.3	2003	
42	66	-92	8.4	772	
51	66	146	4.9	715	

Table 1. Corner merits for the candidate corner points shown in Figure 7 (middle: p=.7). X,Y are the coordinates of the point, T is the rate of change of gradient direction, G is the gradient magnitude, and C=GT.

10	79	-162	10.9	1765	
22	79	35	14.2	497	
32	79	22	5.8	127	*
34	79	250	6.1	1525	
37	79	236	8.6	2029	
42	79	64	6.9	441	
51	79	19	9.3	176	
60	79	-105	11.1	1165	
62	79	221	9.5	2099	
10	84	-37	19.2	710	
17	84	24	8.5	204	*
32	84	298	2.1	625	
34	84	68	6.9	469	
37	84	32	9.9	316	
42	84	71	11.0	781	
51	84	-46	10.2	469	
60	84	-2654	.2	530	
62	84	-129	8.9	1148	
32	91	-131	20.5	2685	
34	91	-37	13.1	484	
37	91	5	4.5	22	*
32	103	-2428	.6	1456	
34	103	-71	7.1	504	
37	103	-524	3.3	1729	
42	103	-29	12.5	362	*
22	116	-301	6.8	2046	
42	116	271	8.4	2276	
51	116	-50	13.7	685	
60	116	-88	4.6	404	*
22	122	-259	3.5	906	
51	122	-69	12.3	848	
62	122	47	16.2	761	*
999					

Table 1, cont'd.

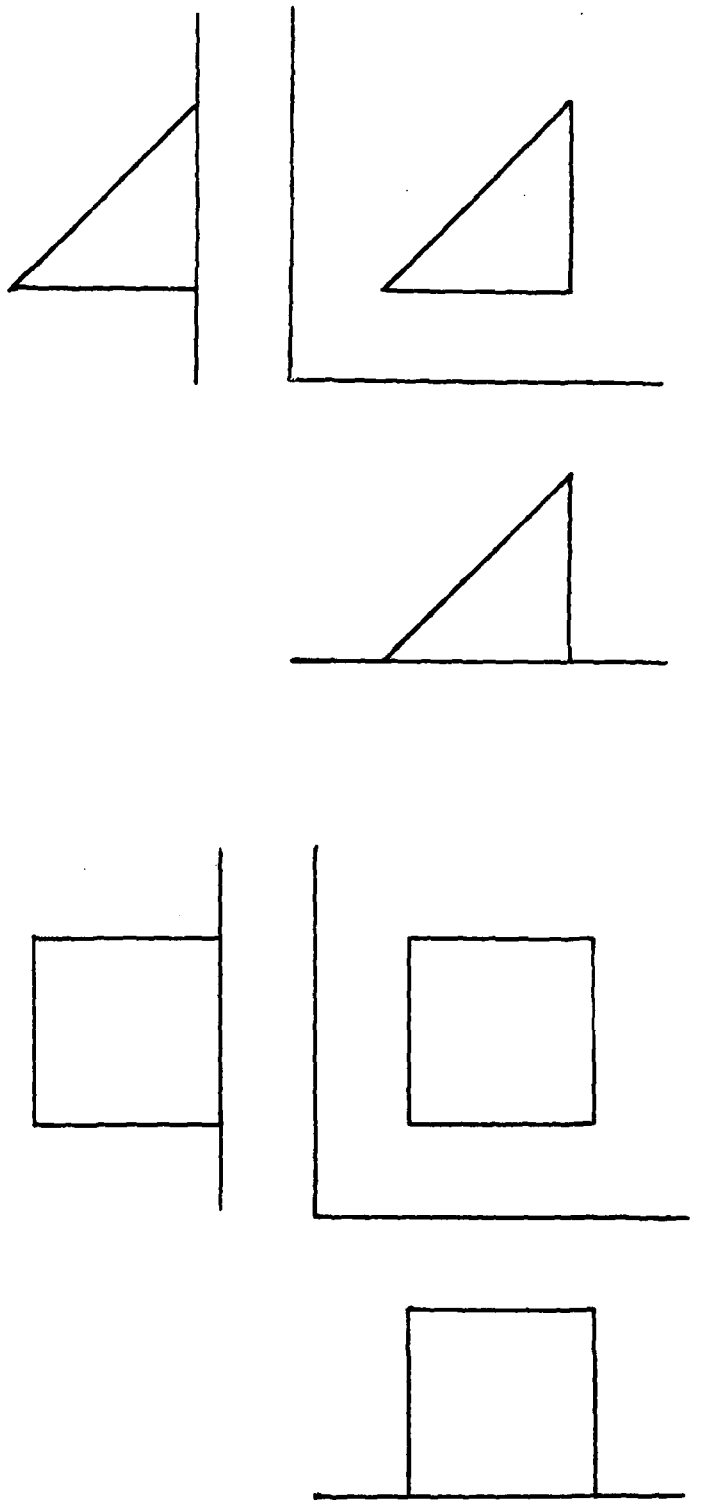


Figure 1. Simple examples of projections; the object has nonzero gray level, the background has level zero.

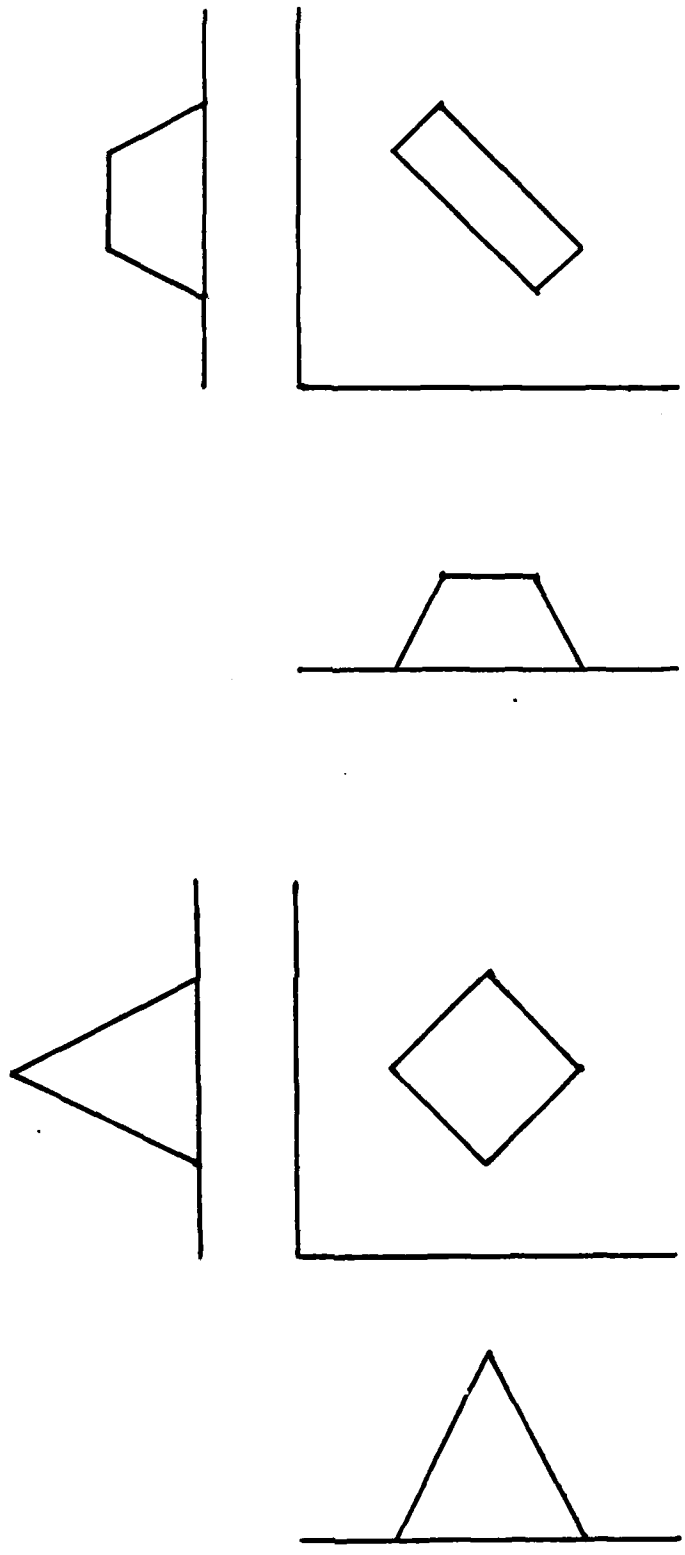


Figure 1, cont'd.

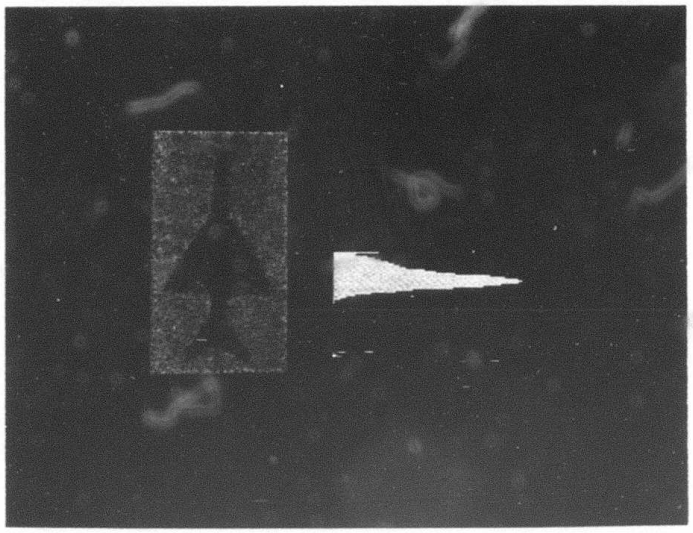


Figure 2. Noisy airplane image and its histogram.

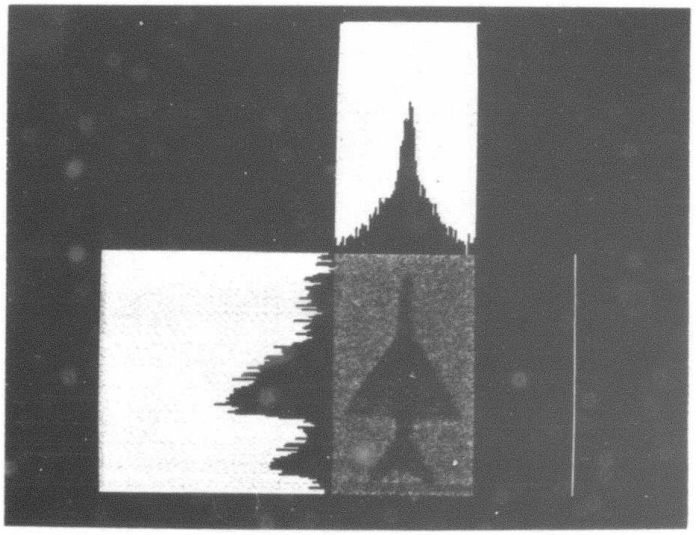
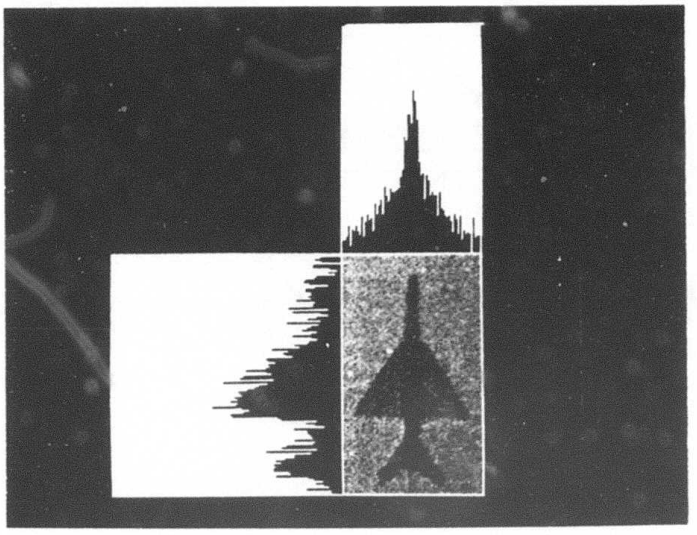
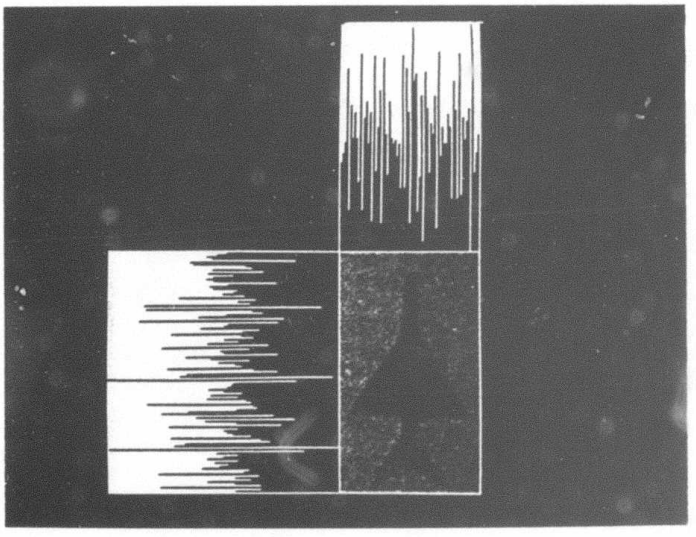


Figure 3. x and y projections of the image.



(a)



(b)

Figure 4. Result of filtering the projections using (a) the truncated Shepp-Logan filter, (b) the one-dimensional digital Laplacian.

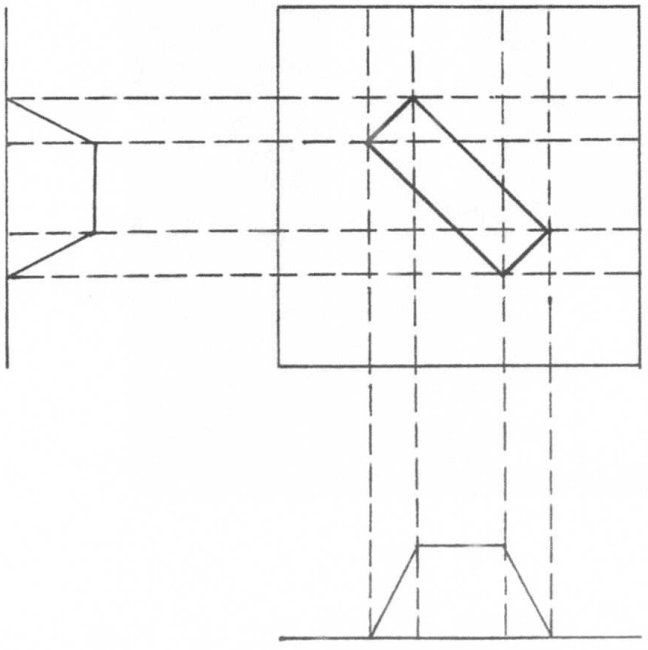


Figure 5. If there are m(n) direction changes on the projections, there are mn possible corner positions in the image. In this simple illustration, m=n=4.

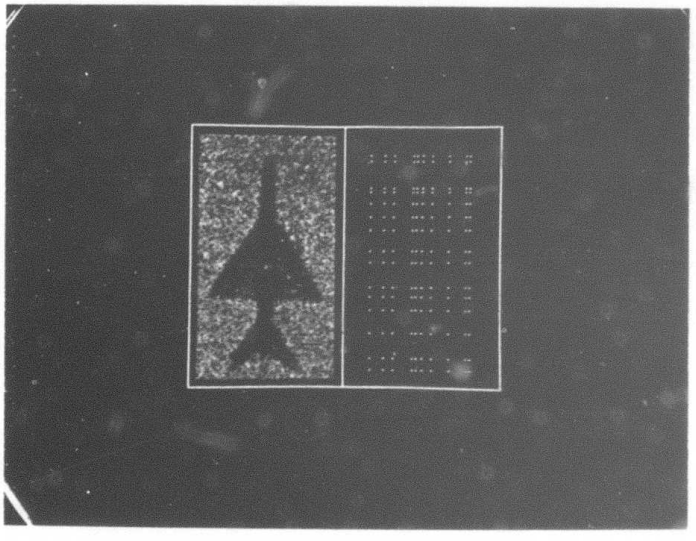


Figure 6. Possible corner positions (165) for the airplane image.

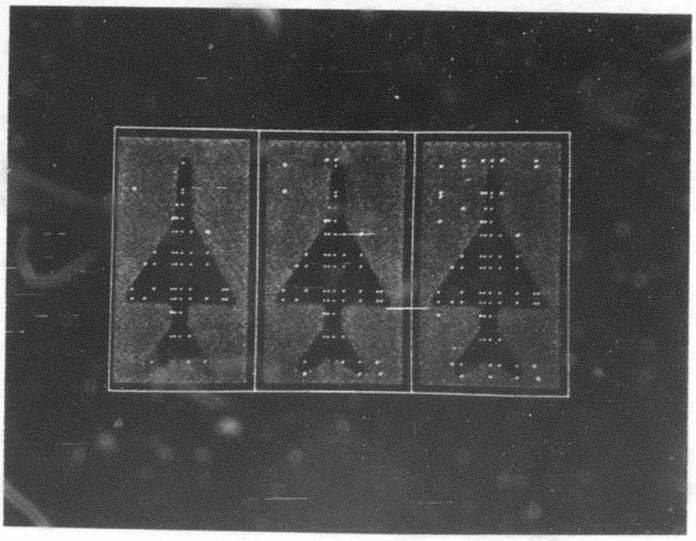


Figure 7. Candidate corner points at which $r \geq p\bar{g}$ for p=.8 (left), .7 (center), and .6 (right), shown superimposed on the original image.

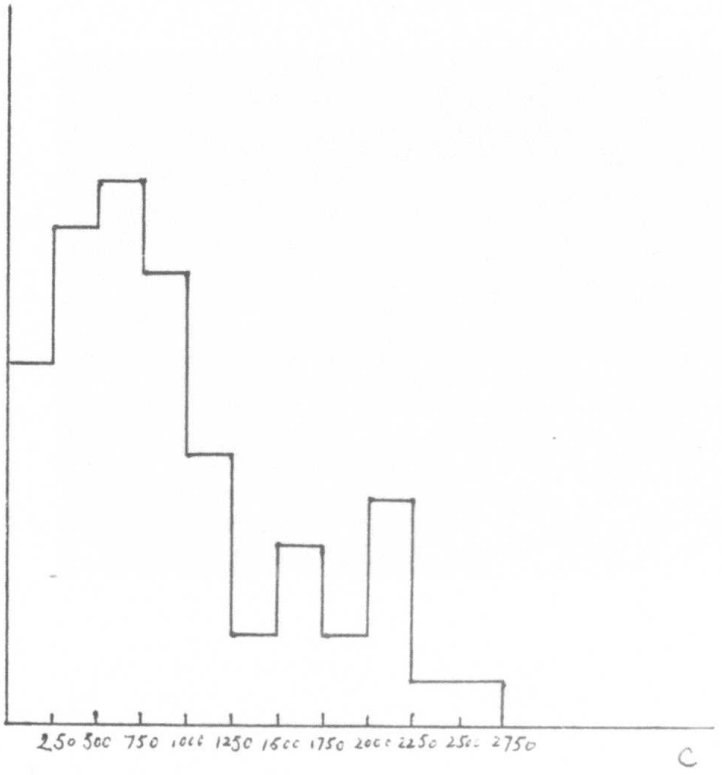


Figure 8. Histogram of corner merits (C=GT) for the candidate points using p=.7.

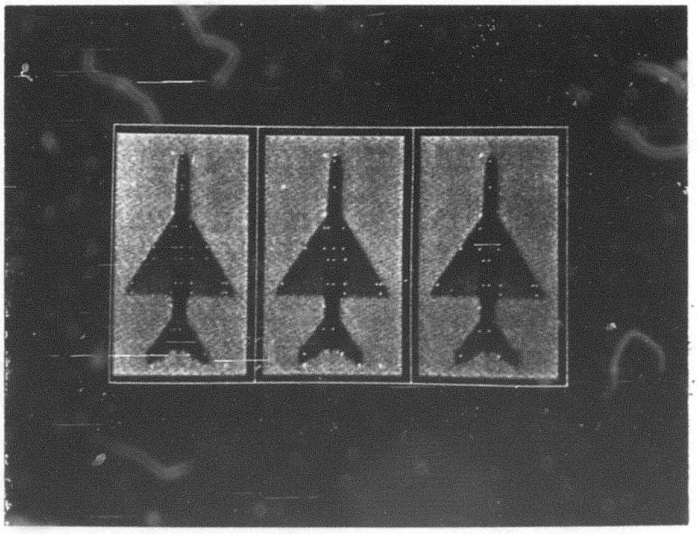


Figure 9. Results of thresholding C at 500 (left), 750 (middle), and 1000 (right) when we used p=.7.

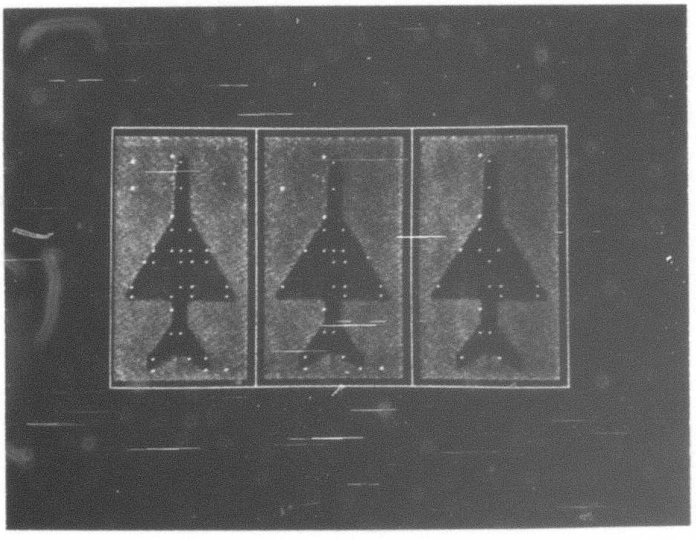


Figure 10. Results of nonmaximum suppression, using a 7×7 neighborhood, applied to Figure 9.

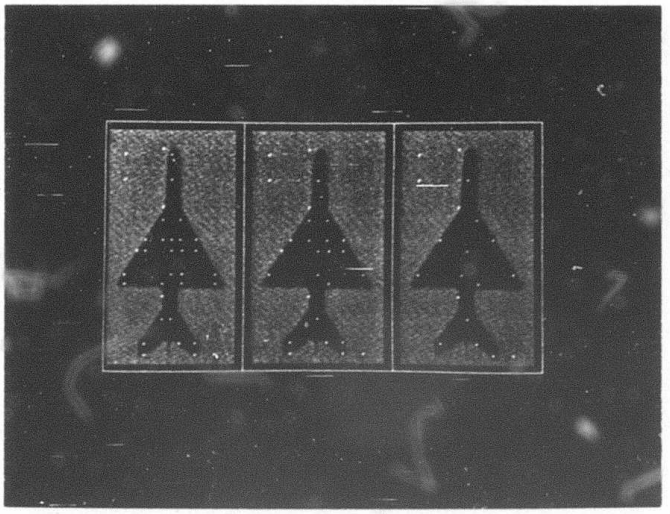


Figure 11. Results of nonmaximum suppression after thresholding at 500, using neighborhoods of 7×7 (left), 11×11 (middle), and 15×15 (right).

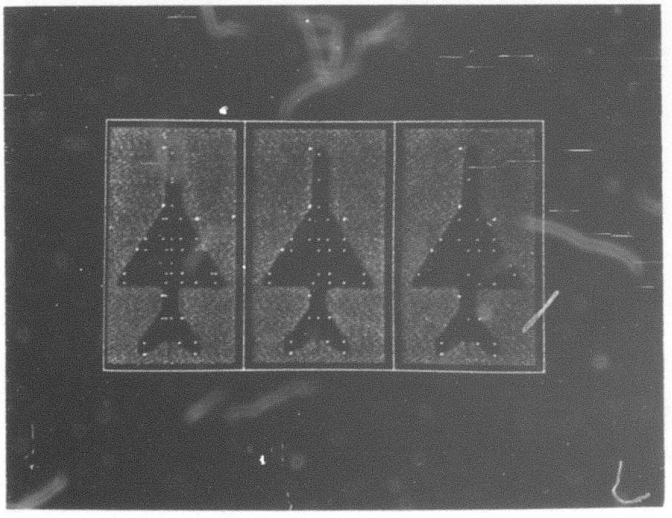
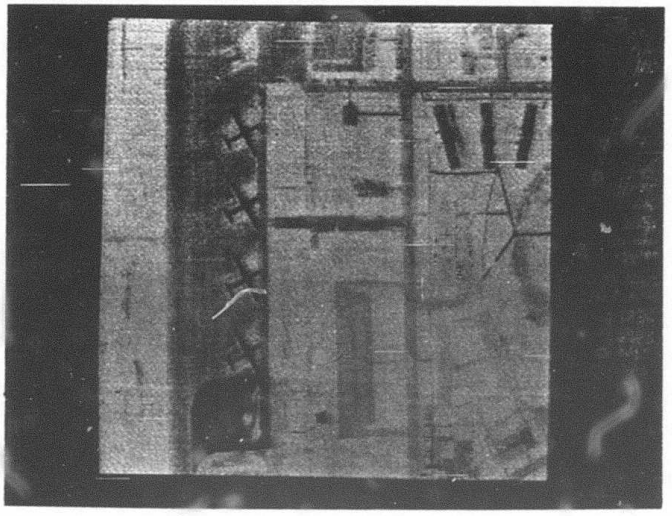
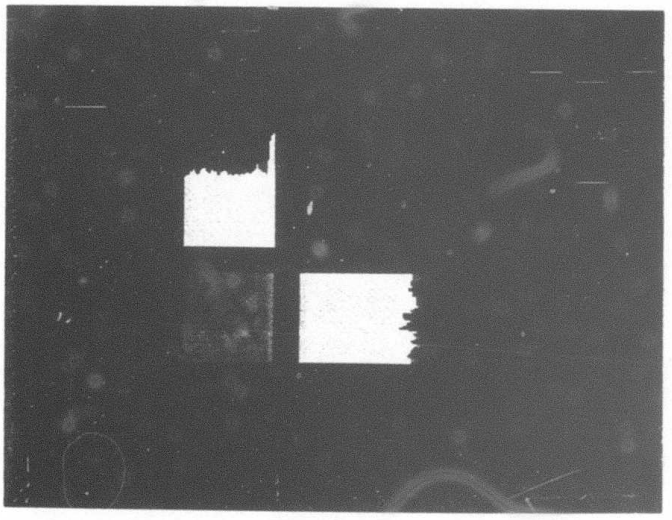


Figure 12. Results of nonmaximum suppression after eliminating the weakest corner in each row and column (starred entries in Table 1), using neighborhoods of 3×3 (left), 7×7 (middle), and 11×11 (right).

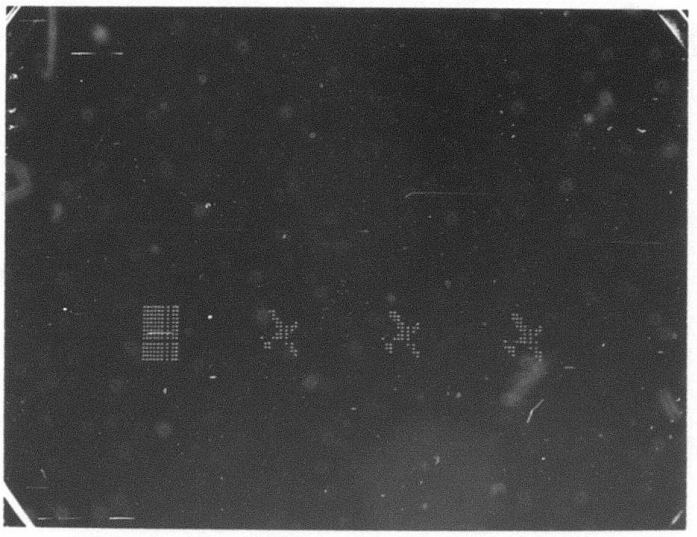


(a)

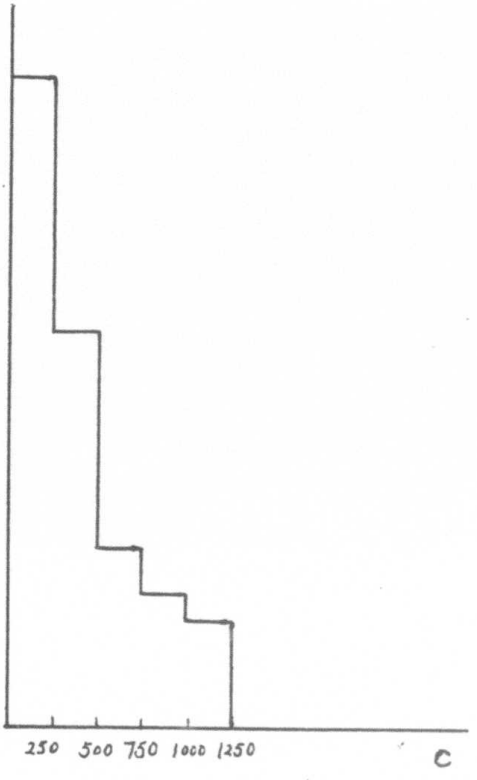


(b)

Figure 13. Another example.
(a) Airport scene; (b) window containing airplane, and its x and y projections.



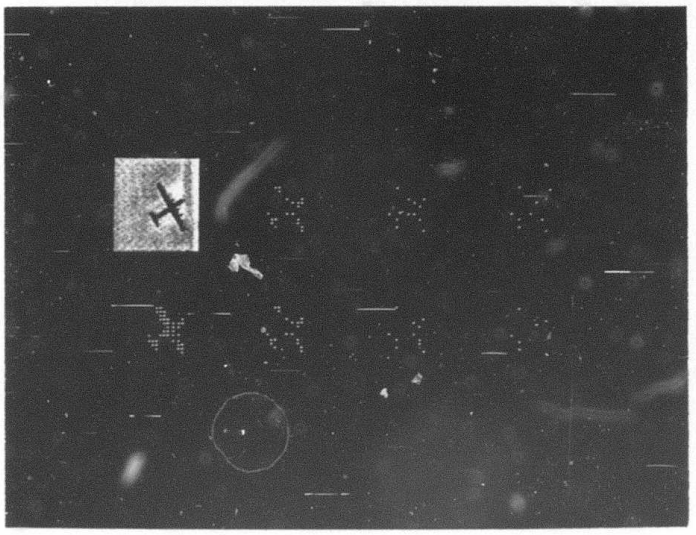
(c)



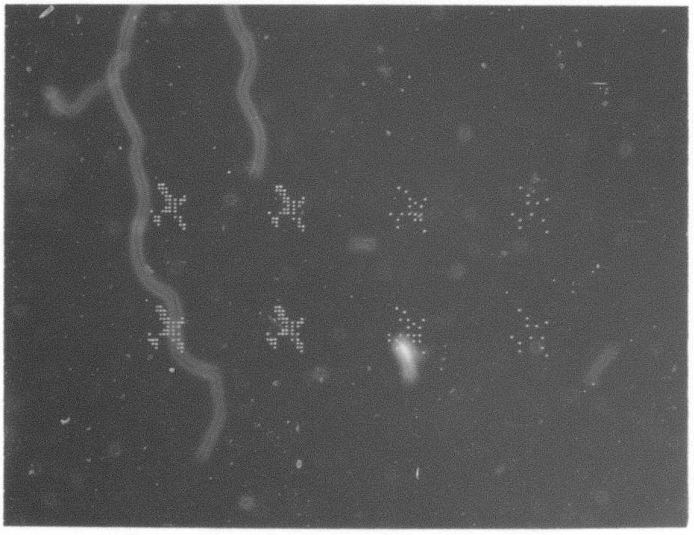
(d)

Figure 13, cont'd. (c) Candidate corner points, and results of applying nonuniformity test $r \geq p\bar{g}$ for p= .8,.7, and .6.

(d) Histogram of corner merit values for p=.6.



(e)



(f)

Figure 13, cont'd: (e) Left column: airplane window and candidate points using p=.6 (same as rightmost output in Figure 13c). Bottom row (cols. 2-4): results of thresholding the corner merit values at 250, 500, and 750. Top row: col. 2 same as bottom row; cols. 3-4, results of nonmaximum suppression using 5×5 and 7×7 neighborhoods. (f) Left column: candidate points using p=.6(bottom) and .7(top) (same as two rightmost outputs in Figure 13c). Remaining columns: results of nonmaximum suppression using 3×3, 5×5, and 7×7 neighborhoods.

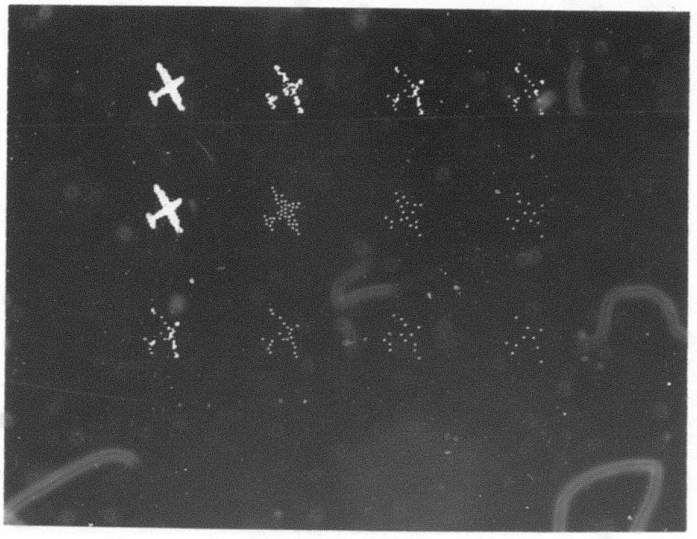


Figure 14. Results of applying the nonuniformity test at every point of the image. Top row: Points for which $r \geq .7\bar{g}$ (left); results of thresholding C at 250,500, and 700. Bottom row: Results of non-maximum suppression without thresholding. Middle row: Results of thresholding at 500 and nonmaximum suppression using 3×3, 5×5, and 7×7 neighborhoods.

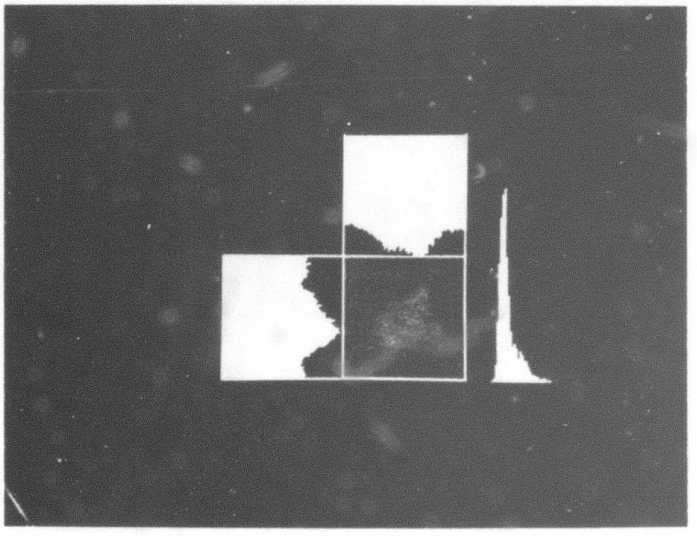
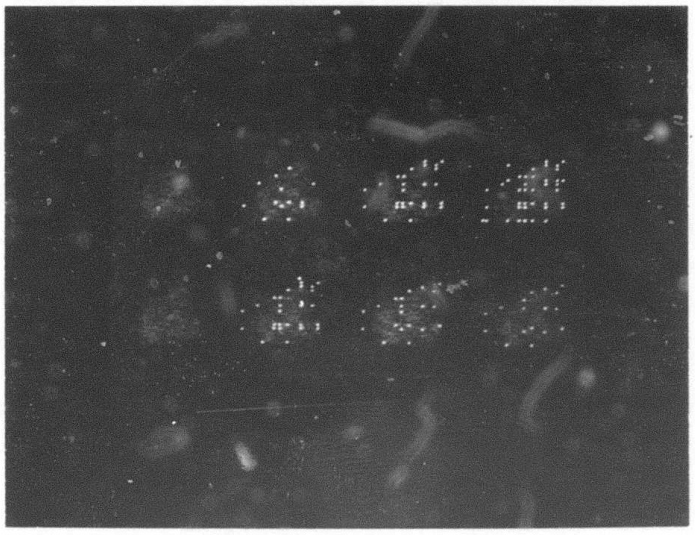
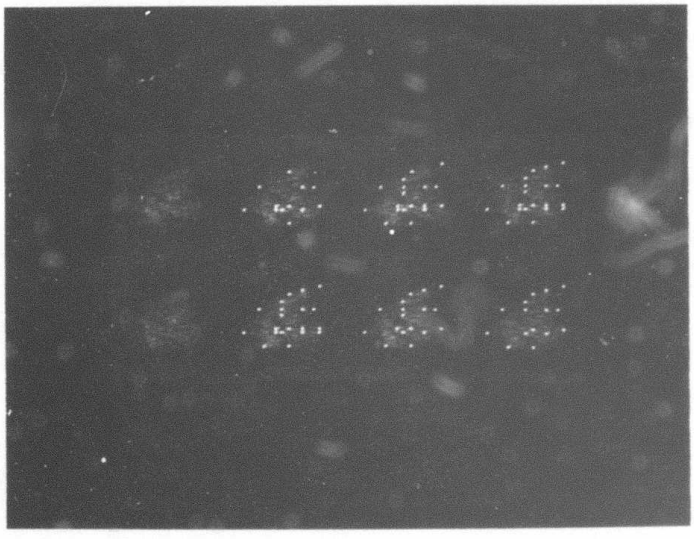


Figure 15(a): Infrared image of a tank and its x and y projections, as well as its histogram.



(b)



(c)

Figure 15(cont'd): (b) Top row: Points for which $r \geq p\bar{g}$ for p=.30, .28, and .25. Bottom row: Points for which $r \geq .28\bar{g}$, and results of nonmaximum suppression using 5×5 and 7×7 neighborhoods. (c) Top row: Results of thresholding the corner merit, for points at which $r \geq .28\bar{g}$, at 200,150, and 100. Bottom row: After thresholding at 100, results of nonmaximum suppression using 3×3, 5×5, and 7×7 neighborhoods.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) FILTERED PROJECTIONS AS AN AID IN CORNER DETECTION	5. TYPE OF REPORT & PERIOD COVERED Technical
	6. PERFORMING ORG. REPORT NUMBER TR-1078
7. AUTHOR(s) Zhong-Quan Wu Azriel Rosenfeld	8. CONTRACT OR GRANT NUMBER(s) DAAG-53-76C-0138
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer Vision Laboratory Computer Science Center University of Maryland College Park, MD 20742	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Night Vision Laboratory Ft. Belvoir, VA 22060	12. REPORT DATE July 1981
	13. NUMBER OF PAGES 27
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Image processing
Pattern recognition
Projections
Corner detection

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Corners are very useful features for such purposes as image matching or shape anlysis, but corner detection is a relatively expensive operation. This paper uses filtered x and y projections, applied to an image containing an object that has not been explicitly segmented from its background, to determine possible positions of corners, so that corner detection can be applied only in the vicinity of these positions. Even in cases where the object would be hard to segment (unimodal histogram), this approach yields a good set of

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

possible cornner positions.